ME 599-5 ARTIFICIAL NEURAL NETWORK
: THEORY AND APPLICATIONS

Professor : Hyungsuck Cho

MLP Neural Network simulation using Matlab

The process to train and test a designed MLP neural network :

1) We make training patterns and test patterns.
2) A network architecture should be defined by **newff** MATLAB function with the number of layers, neurons and transfer functions.
3) The defined neural network architecture is trained by **train** MATLAB function with input patterns and training parameters.
4) We can easily check the result by using a **sim** MATLAB function.

Example :

1. Training Patterns

Input(x) : [-15 -10 -5 0 5 10 15]

Desired Output(y) : function $y = 0.05x^3 - 0.2x^2 - 3x + 20$ for input(x)

[-148.7500 -20.0000 23.7500 20.0000 6.2500 20.0000 98.7500]

2. Architecture of MLP

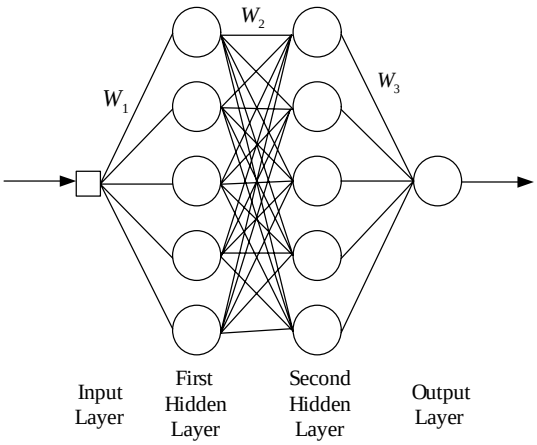

Figure 1. Archtecture of MLP

3. Test Patterns

Input(x) : values from -15 to 15

Desired Output(y) : function $y = 0.05x^3 - 0.2x^2 - 3x + 20$

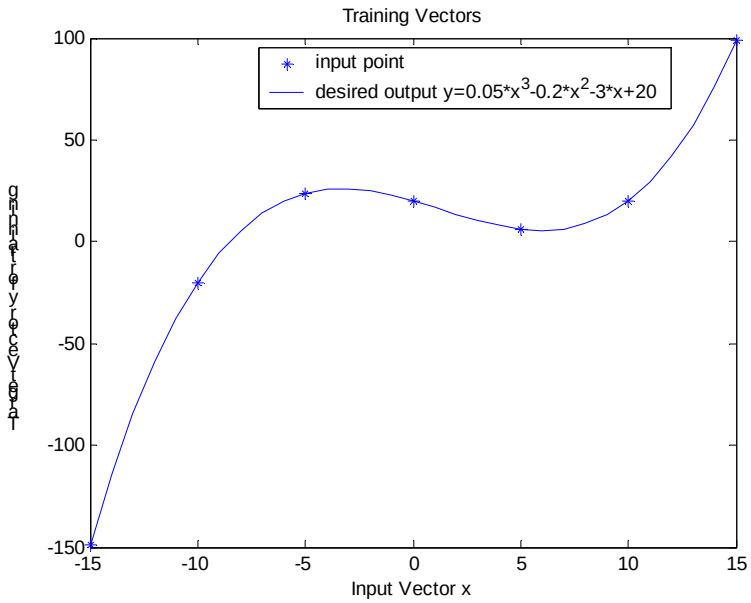

Figure 2. input data

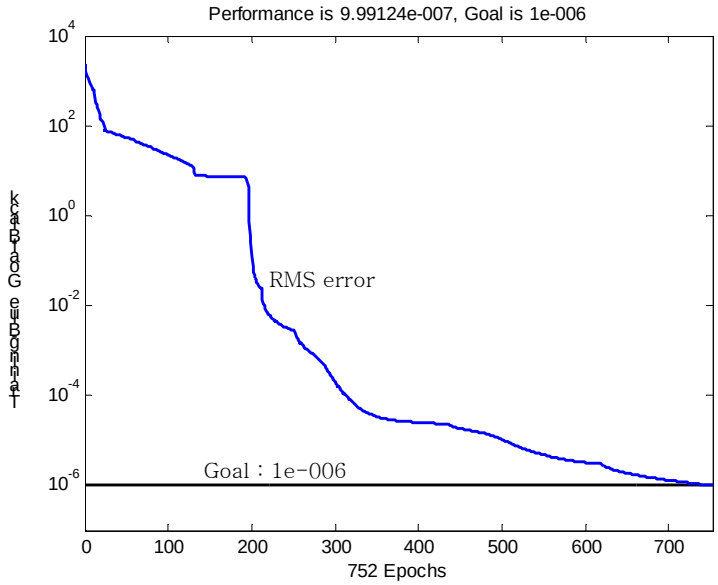

Figure 3. RMS error values at each iterations

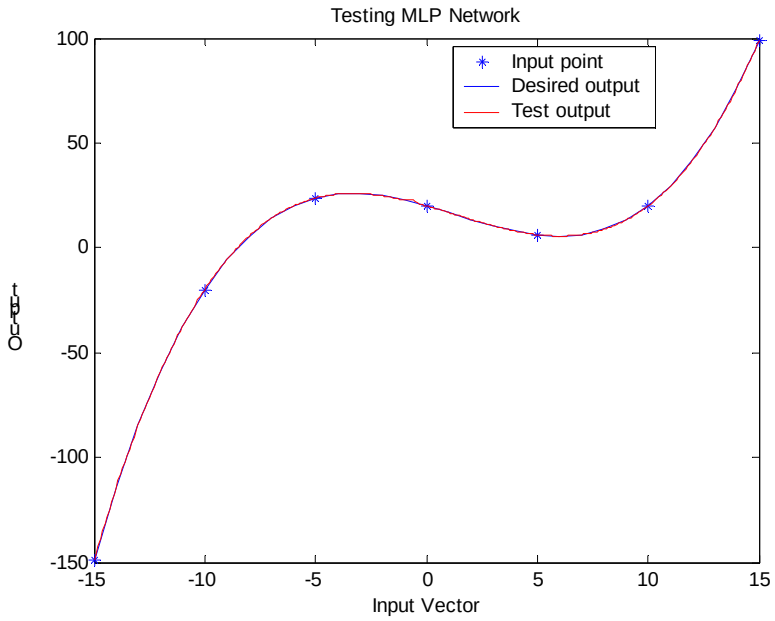

Figure 4. Test pattern simulation

Matlab Program with toolbox :

```
clear all          % clear all data and initializing

% process of making a data set for MLP training
%-------------------------------------------------
x=[-15 -10 -5 0 5 10 15];      % input for training
y=0.05*x.^3-0.2*x.^2-3*x+ 20;     % output for training
%-------------------------------------------------

% Process of drawing the picture of trained MLP
%-------------------------------------------------
x1=[-15:1:15];
y1=0.05*x1.^3-0.2*x1.^2-3*x1+ 20;
figure(1);                    % making a new figure window
plot(x,y,'b*',x1,y1,'b')      % drawing the graph (x,y), (x1,y1)
title('Training Vectors');    % a title of figure
xlabel('Input Vector x');     % x label of figure
ylabel('Target Vector y for trainning');     % y label of figure
legend({'input point','desired output y=0.05*x^3-0.2*x^2-3*x+ 20'})
                              % legend of figure
```

```
%-------------------------------------------------

% training the MLP Neural network
%----------------------------------
% using the newff toolbox function
% input range : -15 ~ 15
% number of neurons of first hidden layer : 5, Log-Sigmoid Transfer function
% number of neurons of second hidden layer : 5, Log-Sigmoid Transfer function
% ouput layer : 1 Linear Transfer function
net = newff([-15 15],[5 5 1],{'logsig' 'logsig' 'purelin'});
                                % define the MLP neural network architecture
% ex) ------------------------------------------------
% MLP neural network with 10 neurons at single hidden layer
%       newff([-15 15],[10 1], {'logsig' 'purelin' } )
% MLP neural network with three hidden layers
%       newff([-15 15],[5 5 5 1], {'logsig' 'logsig' 'logsig' 'purelin' } )
%-------------------------------------------------

% Setting the parameters for training
%-------------------------------------------------
net.trainParam.epochs = 10000;
                % Maximum epochs is 10000 ( iteration number of optimization )
net.trainParam.goal = 0.000001;    % Error limit is 0.000001
%-------------------------------------------------

% training the MLP by using a training pattern
%-------------------------------------------------
net = train(net,x1,y1);
                % training the MLP neural network by using a train function
%-------------------------------------------------

% To draw the result making the test pattern
%-------------------------------------------------
Test=[-15:0.1:15];      % making the input of test pattern
Output=sim(net,Test);       % simulate the MLP network by using a   sim function
                % Output is the result of simulation
```

```
%-------------------------------------------------

% drawing the figure of result of simulation
%-------------------------------------------------
figure(2);
plot(x,y,'b*',x1,y1,'b',Test,Output,'r')
title('Testing MLP Network');
xlabel('Input Vector');
ylabel('Output');
legend({'Input point','Desired output','Test output'})
%-------------------------------------------------
```